Universitat de Girona
Escola Politècnica Superior

Pràctiques ESII

Curs 2023/24

GEINF- GDDV

Pràctica 2

Grup X
Aniol Juanola, Jordi Badia i Guillem Vidal

(October 31, 2023)

Contents

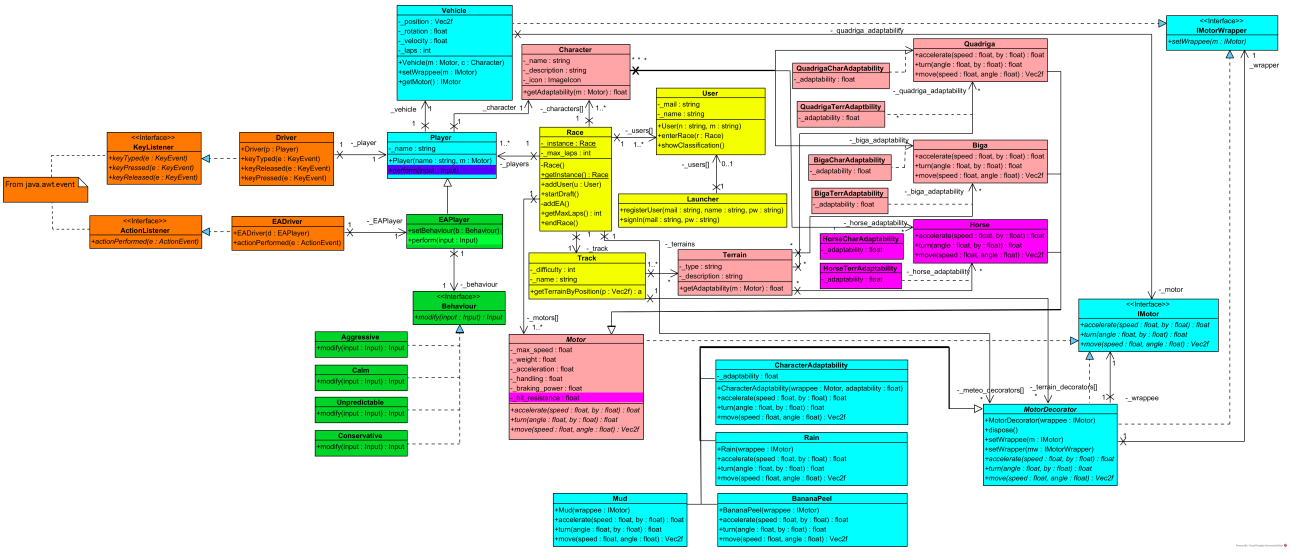
1 Strategy pattern analysis

An obvious use case for the Strategy pattern is the implementation of EA's behaviour. The basic idea is as follows: we have a Driver and a Player for the normal user, and a Driver and a Player for the EA. Each Driver contains a reference to its corresponding Player. EAPlayer extends Player in two ways: it adds a `setBehaviour(b: Behaviour)` method and overrides `perform(input: Input)` such that it calls `modify(input: Input)` before calling its base functionality. See Listing 1.

```
public class Player {

    public void perform(Input input) {

        // Perform operation...

    }

}

public class EAPlayer extends Player {

    private Behaviour _behaviour;

    @Override
    public void perform(Input input) {

        input = _behaviour.modify(input);

        super.perform(input);

    }

    public void setBehaviour(Behaviour behaviour) {

        _behaviour = behaviour;

    }

}
```

Listing 1: Possible implementation of Player.

2 Class diagram



We added the attribute, `_hit_resistance`, and a new class for the vehicle Horse and its corresponding association classes.

3 Strategy pattern example

3.1 Problem description

We have designed and implemented a simple *Strategy* pattern that might be userful in any console application that needs to format text for a desired output. The source code has been designed and compiled with Java 21 in mind, although it is compatible with older versions as well. It is a simple text formatter that, given an input string, it may convert it to different output formats:

Default No modification is done to the input string.

Uppercase The text is converted to UPPERCASE letters.

Binary The binary output of any given string is returned, separating each character with a comma.

Cowsay An ASCII cow is printed with the given string inside a speech bubble. This would be the most 'complicated' output, showcasing the possibilities of this pattern.

3.2 Class diagram